Separation Oracle

Given $x \in \mathbb{R}^n$, assert $x \in P$ or return a separating hyperplane.

Theorem. Let $P \subseteq \mathbb{R}^d$. Assume that $P$ is full-dimensional. There exists an algorithm that finds a hyperplane separating $x$ from $P$.

Min-$s$ Cut

Given digraph $G = (V, E)$, distinguished vertices $s$ and $t$, and edge costs $c_e$, find a minimum weight set of edges that separates every $s$-$t$ path.

Separation Oracle. Shortest $s$-$t$ path with weights $c_e$.

Min Cost Arborescence

Min cost arborescence. Given digraph $G = (V, E)$ distinguished vertices $s$ and $t$, and edge costs $c_e$. Find a subtree of $G$ that contains a directed path from $t$ to all other vertices.

Ellipsoid and Combinatorial Optimization

Cutset-Chvátal-Schrijver. Polytime algorithms for:
- Network synthesis.
- Matroid intersection.
- Chinese postman problem.
- Min weight perfect matching.
- Minimize submodular set function.
- Stability number of a perfect graph.
- Covering of directed cuts of a digraph.

Totally Unimodular Matrices

Def. A matrix $A$ is totally unimodular if every square submatrix is unimodular.

Matrix Games

Matrix game for $n \times m$-dimensional matrices. Given matrices $A = (a_{ij})$ and $B = (b_{ij})$, find the optimal strategies for players $A$ and $B$.

Totally Unimodular Matrices

Theorem. If $A$ is totally unimodular, then every square submatrix of $A$ is unimodular.

Assignment Problem

Assignment problem. Assign $n$ tasks to $m$ workers to minimize total cost, where $c_{ij}$ is cost of assigning task $i$ to worker $j$.

Matrix Games

Matrix game for $n \times m$-dimensional matrices. Given matrices $A$ and $B$, find the optimal strategies for players $A$ and $B$.

Linear Programming III

Linear programming problems. Minimize a linear function subject to linear constraints.

Assignment Problem: LP Formulation

Assignment problem. Minimize total cost subject to assignment constraints.

Matrix Games

Matrix game for $n \times m$-dimensional matrices. Given matrices $A$ and $B$, find the optimal strategies for players $A$ and $B$.

Matrix Games

Matrix game for $n \times m$-dimensional matrices. Given matrices $A$ and $B$, find the optimal strategies for players $A$ and $B$.

Linear Programming III

Linear programming problems. Minimize a linear function subject to linear constraints.

Assignment Problem: LP Formulation

Assignment problem. Minimize total cost subject to assignment constraints.

Matrix Games

Matrix game for $n \times m$-dimensional matrices. Given matrices $A$ and $B$, find the optimal strategies for players $A$ and $B$.
**Minimax Theorem**

**Theorem.** [von Neumann 1928] For every $A \in \mathbb{R}^{m \times n}$,

$$\min_{x} \max_{y} \langle A \cdot x, y \rangle = \max_{y} \min_{x} \langle A \cdot x, y \rangle \quad (*).$$

This is called the **minimax theorem**.

**Consequence.** As long as your mixed strategy is optimal, you can reveal it to your opponent.

**Theorem.** Nash equilibrium exist for 2-person zero-sum games. Moreover, they are poly-time computable.

**Kuhn's simplified poker.**

- Deck of 3 cards, numbered 1, 2, and 3.
- Each player antes 1.
- One round of betting (1 bet).
- If pass-pass, pass-bet, or bet-bet, player with higher card wins; otherwise player that bets wins.

**Kuhn's strategies for X.**
1. Pass.
2. Pass; if $Y$ bets, bet.

**Kuhn's strategies for Y.**
1. Pass no matter what $X$ did.
2. If $X$ passes, pass; if $X$ bets, bet.
3. If $X$ passes, bet; if $Y$ bets, pass.
4. Bet no matter what $X$ did.

**Application: Poker**

**Optimal strategy for $X$.**
- When dealt 1, mix strategies 1 and 3 in ratio 1:1.
- When dealt 2, mix strategies 1 and 2 in ratio 1:1.
- When dealt 3, mix strategies 2 and 3 in ratio 1:1.

**Optimal strategy for $Y$.**
- When dealt 1, mix strategies 1 and 3 in ratio 2:1.
- When dealt 2, mix strategies 1 and 2 in ratio 2:1.
- When dealt 3, use strategy 4.

**Value of game.** $-1/2$ for $X$.

**Gambling lessons.** Optimal strategies involve bluffing and trapping. Player who acts last has advantage.

---

**Linear Programming III**

**Strongly-Polynomial**

An algorithm is strongly polynomial if:
- Elementary ops: $+$, $-$, $/$, comparison
- # ops is polynomial in the dimension of input.
- Polynomial space on a classic TM.

Ex. Merge-sort: $O(n \log n)$. 
Ex. Edmonds-Karp max-flow: $O(n^4)$. 
Ex. Gaussian elimination: $O(n^3)$ arithmetic ops.

Ex. Ellipsoid: $O(n^4)$ arithmetic ops. 
Ex. Interior point method: $O(n^3)$ arithmetic ops.

Open problem. Strongly polynomial algorithms for LP? 
Open problem. Is LP in $P$?

---

**New York Times Article**

An Approach to Difficult Problems

- Ellipsoid algorithm
- Combinatorial optimization
- Matrix games
- Open problems